I/O

April- June 2014

A quarterly newsletter for LANL's HPC user community

In this issue

Things you should know........................1

Consultants Corner...................2

Software and Tool News.........................4

Machines News.....................6

HPC-Behind the Scenes......................8

Quarterly Stats.......12

Current Machines- a snapshot in time.......13

LA-UR-14-25322

Things you should know...........

Standard service features

The Integrated Computing Network (ICN) Consulting Team provides user support on a wide variety of HPC topics:

- Programming, languages, debugging
- Parallel computing
- HPC and Unix operating systems, utilities, libraries
- Unix / Linux scripting
- Archival storage: High Performance Storage System (HPSS), General Parallel File System (GPFS)
- Desktop backup (TSM), storage, file transfer, network
- Mercury: Cross-network file transfer service
- HPC infrastructure in both the Open and Secure

Service hours

Monday through Friday, 8 a.m. - 12 p.m. and 1 p.m. - 5 p.m

After-hours/Urgent support-can transfer to 7x24 operations desk

Phone #

505-665-4444 opt 3

Email

consult@lanl.gov

Documentation

http://hpc.lanl.gov

HPC Change Control Calendar

http://ccp.lanl.gov

Consultants Corner

Application optimization guidelines
Our Linux-based HPC platforms are designed to run large-scale scientific applications. We configured them to segregate different types of usage in order to optimize user application workload on the compute nodes. As you prepare and execute your application, please observe a few simple guidelines for a smooth running cluster.

Guidelines
1. Use the front-end nodes for a limited set of activity:

- text **editing**,
- job script **set-up,**
- pathname arrangements,
- **compilations**,
- **linking**, and
- **job launching.**

Even large parallel compilations can tie-up a front-end; you are better off running those on a compute node. Avoid running the tar command on a large subdirectory, a file transfer (scp, rsync, etc.), or running an application for more than 10 minutes.

2. Use the compute nodes for running your apps and performing large parallel compilations.

In most cases, the HPC clusters have two major components: front-end node(s) and back-end compute nodes. A front-end is small, usually identical to one cluster compute node, and shared by all registered users for that platform. It has no interconnect for parallel code and cannot perform parallel I/O operations. Even just a few serial applications can tie-up memory and I/O bandwidth, but it's the only place we allow node-sharing.

Think of it this way: you share the front-end node with hundreds of people doing the same thing you are doing. The front-end node typically has between 100 and 500 user accounts. Although not everyone logs-in every day, the active front-end nodes usually see between 30 and 160 unique users logged-in who collectively run up to 200 individual shells.

For example, a quick snapshot of the YeTI side of Pinto shows 33 users with 42 individual login shells:

```
pi-fey> w | cut -c1-9 | uniq | wc
   35     35    350
pi-fey> w | wc
   44    367   3185
pi-fey>
```

As you can imagine, if every user decided to run a serial application or a file transfer on a front-end node simultaneously, it would crash. In fact, periodically we have been forced to reboot front-end nodes at times when they appear locked-up from excessive CPU or I/O load.

On most of our ICN Clusters it is possible to ignore our usage model guidelines. We do not expend a lot of effort in policing the clusters, and we do not always stop users from being bad citizens. We will call you if we receive a report of your running outside the model, and occasionally we will spot-check. For the most part we are using the honor system for users to stick with this model to benefit everyone. Please do your part by following the guidelines.

With this usage model, applications run on dedicated back-end processors with minimal interference from system daemons, other user activity, network traffic, unrelated I/O traffic, etc. We do not allow sharing of back-end compute nodes -- you have exclusive access to them from Moab (ie. the msub command) and can perform whatever tasks you wish on your allocated nodes. For more information, see:
Usage Model -- http://hpc.lanl.gov/usage_model
Examples of Submitting Jobs --
http://hpc.lanl.gov/moab_example
Use FTAs for moving data --
http://hpc.lanl.gov/fta_home

NEW: File Touching Policy

As of this writing, we have not published a policy to prohibit file-touching. Many users touch files to avoid purging of our scratch space, and this adversely affects I/O performance and application reliability. We are one of the few large scientific computing centers that lacks a policy, but this is changing soon. We will notify users when we have written a new policy and begin enforcing it.

If you need to preserve data, you can archive it offline to HPSS or GPFS (Turquoise) and retrieve the data on an as-needed basis. See:
http://hpss-info.lanl.gov/hpss/Index.php
http://hpc.lanl.gov/fta_home
http://hpc.lanl.gov/turquoise_archive

Touching files is both inappropriate and inconsiderate to other users of the shared scratch filesystems. While not yet explicitly forbidden by policy, file-touching clearly falls into the category of "unauthorized or improper use" since it deliberately circumvents our established Purge Policy -- http://hpc.lanl.gov/purge_policy

As always, if you are not able to answer your question to your satisfaction, please contact ICN Consulting: consult@lanl.gov, or 505-665-4444 option 3.

Consultants
left to right, back to front
Ben Santos, Hal Marshall, Riley Arnaudville, Rob Derrick
Giovanni Cone, Rob Cunningham
David Kratzer

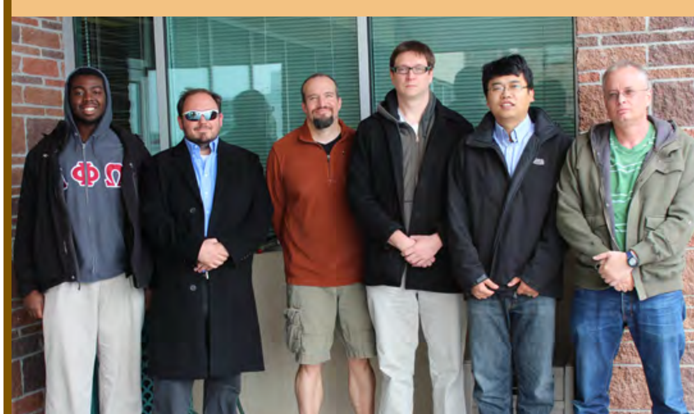
Ultrascale Systems Research Center

The Ultrascale Systems Research Center (USCR) is a colaboration between the New Mexico Consortium and LANL to engage universities and industry nationally in support of exascale research. USRC investigates the challenges of computing at extreme scales - millions rather than thousands of computers. Such large systems pose questions that have not yet been answered. USRC research includes the following research topics as they relate to exascale computing:

OS/systems/network software stacks
Scalable and reliable runtimes and middleware
IO/Storage, parallel file systems
Data Intensive (DISC)
Cyber-security

USRC researchers work on challenges such as developing fault-tolerant scalable system services and understanding how interconnects, operating systems, and other factors effect performance as you drastically increase scale. Resilience research draws ideas from disciplines ranging from biology to social networks to build systems that can adapt to failures. Network research investigates typologies, routing protocols, switching techniques, flow control, and congestion control to address current issues known to cause problems. Understanding how to query enormous sets of scientific data is another critical task as well as investigating storage and data retrieval at exascale.

https://www.newmexicoconsortium.org/research/advanced-computing/usrc



Software and Tool News

Programming Environments and Runtime Team Announcements:

Summer Tools Workshops:

TAU – Tuning and Analysis Utilities

July 31st -A portable profiling and tracing toolkit for performance analysis of parallel C, Fortran, C++, UPC, Java and Python, will be at LANL conducting a free user workshop

Open|Speedshop

August 19th- The development team will be onsite to conduct a user workshop and demonstrate new features of Open|Speedshop Performance Analysis tool

To coincide with the workshops, new versions will be made available on the clusters as required.

Allinea DDT+MAP unified tokens upgrade

You spoke, we listened! In the next few weeks, expect Allinea MAP to be available on all of our production clusters. The upgrade features unified tokens, where the total (increased) Allinea tokens are interchangeable between tools. Allinea DDT has proven beneficial to code teams when other available products proved tedious or ineffective, and the intuitive QT4 based GUIs for both products make debugging and analyzing code almost enjoyable! Look forward to new trials of the Performance Reports product soon on our clusters.

Programming Environments and Runtime Team
left to right
David Gunter, Riley Arnaudville, Jennifer Green, David Shrader, Giovanni Cone, Marti Hill, and Jorge Roman

New Software

- IDL 8.3 now available on all production clusters
- OpenMPI default moved to 1.6.5
- TMux
- Intel Cluster Studio XE 2013
- PGI/14.4 available in the open

Product Spotlight: Tmux
http://hpc.lanl.gov/tmux

Using Terminal Multiplexers for Multitasking and Session Persistence

Terminal multiplexers offer the ability to have multiple virtual-terminals over a **single SSH session** and provide the ability to have a persistent environment by disconnecting from a multiplexer and reconnecting later. The multiplexers also provide persistence in the event of a network disconnect or a session timeout (Turquoise environment).

The tmux terminal multiplexer application is not provided in the default packages of the TOSS/CHAOS Linux distribution. A modulefile named "tmux" is listed in the "tools" category and is available for loading on our HPC platforms. Simply issue the command `module load tmux` to have access to this particular terminal multiplexer. The default tmux session, with no user defined configuration file, is a bit more enlightening than the GNU Screen default. As the following screenshot shows, a default tmux session includes an informative status bar in green.

```
cj-fe2:1025  17:42:53
$ tmux list-sess
DEVEL: 6 windows (created Mon Jan 27 17:48:55 2014) [80x24]
SAMPLE: 3 windows (created Mon Jan 27 17:42:53 2014) [80x24] (attached)
TEST: 4 windows (created Mon Jan 27 17:45:54 2014) [80x24]

cj-fe2:1026  17:50:17
$ tmux list-cli
/dev/pts/1: SAMPLE [80x25 xterm-256color]

cj-fe2:1027  17:50:22
$

[SAMPLE] 0:tcsh* 1:tcsh  2:tcsh-              "cj-fe2.lanl.gov" 17:52 27-Jan-14
```

To obtain the tmux sample shown above, the following steps were done:

- module load tmux
- tmux new -s SAMPLE (SAMPLE being a tmux session name) followed by
- **Ctrl+b c** entered several times to get three tmux windows

The Programming Environment and Runtime Team, formerly PTools, of LANL's High Performance Computing Division, strives to provide useful tools and knowledgeable staff to assist users in debugging and optimizing applications on the production clusters. Some of the services they provide include one-on-one developer support, hosting workshops with third-party software vendors, user educational outreach, and software documentation hosted on hpc.lanl.gov

Lustre- HPC's newest scratch file system

HPC Division has fielded a new Lustre-based parallel-distributed file system consisting of 3 PB of object store scratch space. This new file system more than doubles the previous amount of scratch space, increasing our total offering from just under 2 Petabytes to almost 5 Petabytes of storage space. As part of this deployment, LANL personnel modified the design of the internal network in order to compartmentalize the high volume communication paths. This is known as Fine Grain Routing (FGR) and is a relatively new concept within the Lustre community. It allows for more efficient networking and facilitates future scaling of the file system while using less hardware and cabling. The read/write speed on this new system has the potential to perform almost twice as fast as our previous scratch file systems. On our newest compute cluster, Wolf (an Institutional Computing resource), Lustre performance reached 40GB/s, as compared to 24GB/s on the older file systems. Older LANL compute clusters are bound by the IO node structure, but still achieved upwards of 28GB/s with the new file system.

The existing parallel file systems are Panasas-based. In contrast, this new file system uses Lustre at the software level. While Panasas has been a very stable solution for many years, it is a proprietary file system that runs on vendor-specific hardware. Lustre is an open source file system that can run on commodity hardware. To gain familiarity with maintaining, monitoring and tuning a Lustre file system, this first implementation of Lustre was purchased as a vendor-provided solution running on DDN hardware. By doing so, LANL personnel now have the knowledge and experience to allow the option of deploying future Lustre systems on cost-effective, yet still robust, commodity hardware. The Lustre file system has been mounted on LANL computer clusters, including Mustang, Wolf, Moonlight, Pinto, Lightshow, Conejo, and Mapache, as resource support for Advanced Simulation and Computing (ASC) Program projects and activities. This includes the Predictive Science Academic Alliance Program (PSAAP) projects, as well as resource support for open science projects and activities through the Institutional Computing Program.

This file system was implemented, tested and vetted by HPC-3 and HPC-5 personnel, including:

Lustre Team
Front Row: David Sherrill, Susan Coulter, Terri Morris, Bobbie Jo Lind
Second Row: Ben Santos, Alfred Torrez, Satsangat Kahlsa, Terri Bednar, Mike Mason
Missing: Julianne Stidham, Ben McClelland

Machines (coming/going/planned/technologies)

Glome update

Glome is the same size as it was before: it has 96 compute nodes, each with 8 compute cores, giving us a total of 768 cores in the system. Each node has 32 GB of RAM, providing 3TB of RAM across the system, and each node also has 8TB worth of storage. Formatted, this gives the system 687TB of HDFS storage. The whole system is interconnected via Ethernet and InfiniBand, and all of the nodes run the standard tri-lab TOSS operating system. On top of that we are running Apache Hadoop 2 (with MapReduce 2 and YARN), as well as Apache Spark and Mahout. Glome is designed to be a production data intensive workload system.

Meanwhile we have also been bringing up Glome's twin, Kugel. Kugel is the same size as Glome and has the same hardware configuration, but is being used for more experimental data intensive projects. We are currently in active development of a software stack that will run virtual machines on top of KVM and QEMU and will allow jobs to be run in user-defined software environments. Kugel is not open to general users yet, but is under ongoing development.

Users have been running on Glome since the beginning of the year, and both clusters will be used to support proposals accepted through the the IC program's data intensive computing call for proposals.

Tim Randles with Glome

New Campaign Storage System Release in Turquoise Network

HPC Division recently deployed a new parallel data-storage capability in the Turquoise Network to provide a new medium-term storage tier for LANL scientists.

The newly deployed Campaign Storage system is not an archive; it is essentially a scratch file system that allows selected data to be stored up to a year with fast recall. This enables data to be cleared from Scratch space while simultaneously allowing users to keep large data sets close to HPC systems without the "pain" of writing to tape. Campaign Storage enables both simulation and "big data" users to conduct investigations of their data over a longer span of time.

Supercomputer-based simulations can run for many months and generate many petabytes of data. Currently data is stored short term in scratch parallel file systems. Scratch space is not automatically backed up. It is the first location to which large calculations dump data. This allows scientists to examine the data to determine what is important and what should be saved. Data stored on scratch file systems is typically deleted after two weeks. Scientists who wish to retain their data for a longer period write the data from scratch to archive, where it can be stored long-term or even permanently; however, recall performance from archive can be slow.

The recent Campaign Storage deployment, which was funded by LANL's Institutional Computing Program, is a 1.9 petabyte GPFS-based file system that uses commodity disk with Raid controller providing Raid 6 for data protection. It allows data transfer at up to 2 Gigabytes per second (in contrast to typical tape recall rates of 180 Megabytes per second or less). GPFS (General Parallel File System) is a high-performance clustered file system developed by IBM. The LANL Campaign deployment consists of 2 clusters, a GPFS cluster and a Linux IO cluster. The GPFS cluster consists of one master node, 2 GPFS server nodes, and 4 GPFS client nodes (the user accessible area). The Linux IO (LIO) cluster consists of one master node and 18 LIO nodes. This system is a RAID6 deployment on disk. A future deployment is planned that will utilize erasure coded disk, which will allow expansion beyond 2PB of storage.

Campaign storage allocations are granted through the annual Institutional Computing (IC) call for proposal process. Campaign Storage user access expires with the associated IC allocation. The main use case is projects with a need for longer-term access to large data sets that would be negatively impacted by standard purge policies and/or are difficult or impossible to recall in a timely fashion from archive. Campaign Storage is a cost-effective, commodity-disk solution that provides more resources for the invested dollar.

Documentation on Campaign Storage may be found online at the following url: http://hpc.lanl.gov/turquoise_filesystems#campaignstorage

Campaign Storage Development Team
bottom row, (left to right):
Christopher Hoffman, Kyle Lamb, Carolyn Connor, David Sherrill, Brett Kettering
Top row (left to right):
David Bonnie, Daryl Grunau
Not shown: Mike Mason, Susan Coulter, Ben McClelland, Randy Crook, Bob Darlington

A review of LANL's PaScalBB (Parallel and Scalable I/O Back Bone) server I/O network – Past, Present, and Future

PaScalBB (Parallel and Scalable I/O back bone infrastructure, shown in Figure-1) architecture is designed as a highly scalable server I/O network infrastructure to meet constantly increasing computation power and storage bandwidth needs. The main goal of PaScalBB is to provide cost-effective, high performance, efficient, reliable, parallel, and scalable I/O capabilities for scientific applications running on very large-scale clusters. Data-intensive scientific simulation-based analysis normally requires efficient transfer of a huge volume of complex data to large scale storage systems, other local machines for computing/analysis/visualization, and to the WAN for offsite use.

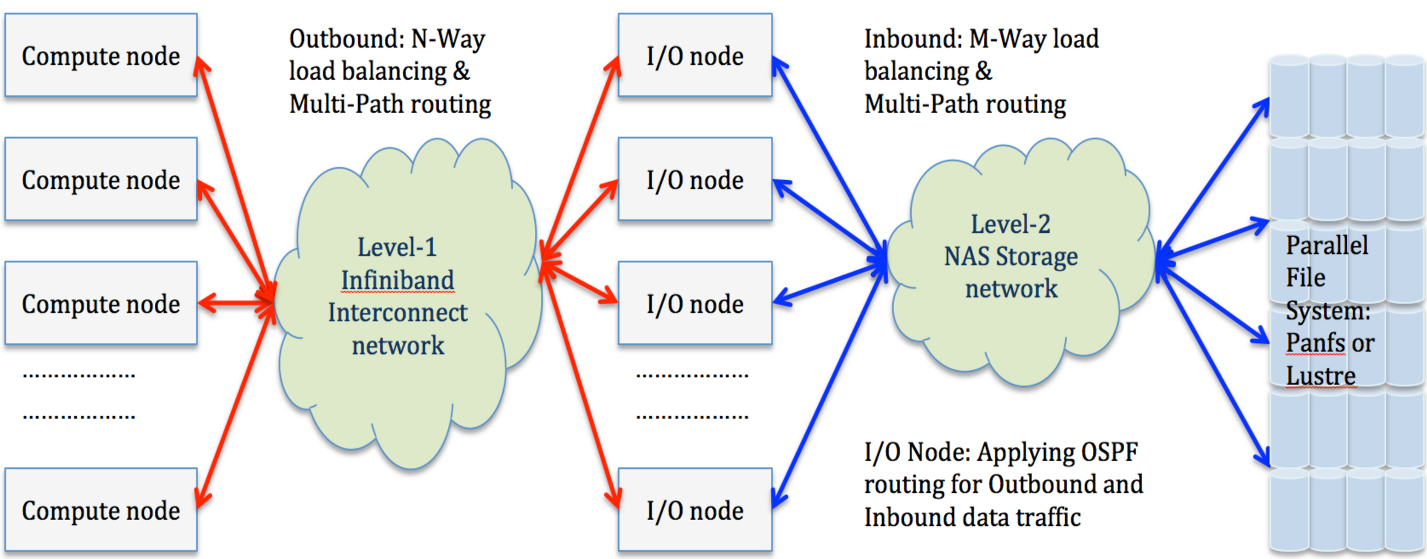


Figure-1: System Diagram for PaScalBB Server I/O architecture

Main features

PaScalBB adopts several scale-up (parallel) and scale-out (scalable) networking features such as:

1) It provides bi-level switch-fabric interconnected systems by combining high speed interconnects for inter-processes message passing requirement and low-cost Gigabit Ethernet interconnect for IP based global storage access.
2) It supports a bandwidth on demand linear scaling I/O network architecture without re-wiring and reconfiguring the system.
3) It adopts Equal Cost Multi-Path (ECMP) and Open Shortest path First (OSPF) routing scheme. Multi-path routing is used to provide balanced outbound traffic to the multiple I/O gateways. It also supports failover and dead-gateway detection capability for choosing good routes from active I/O gateways.
4) It improves reliability through reducing large number of network components in server I/O network, and
5) It supports global storage/file systems in heterogeneous multi-cluster and Grids computing environment.

Self-management and Reliability feature

We also implemented a dependable server I/O fault-management mechanism used in LANL's PaScalBB based High Performance Computing cluster systems to run computational jobs 24x7 without service interruption during an unexpected physical I/O link failures and connection loss. This mechanism, named Dead Server I/O Gateway Detection and Recovery (DGD), can detect a data path connectivity problem within seconds when it happens, removes the entry of a dead I/O gateway from a Multi-Path routing table, transfers connecting I/O path to available entrance in a Multi-Path routing table, and then resumes the existing I/O data stream. The DGD can tolerate multiple single points of failures; keep the streaming I/O data moving, and seamlessly continue and finish computation jobs. The DGD is a Self-management feature. It supports self-healing and auto-reconfiguration.

Scalability feature for supporting multi-clusters environment

With (a) network Layer-2 and Layer-3 fail-over support from Linux kernel routing implementation and Ethernet switch capabilities and (b) a global multicast domain support from scalable metadata servers, the PaScalBB I/O networking Infrastructure can support a global storage system in a heterogeneous multi-cluster environment.

Figure-2 illustrates the top-level view of PaScalBB in a heterogeneous multi-cluster and Grids environment. We can apply PaScalBB to support a heterogeneous multi-clusters environment that consists of several independent large-scale cluster systems. PaScalBB provides an ability to mount a single name space global file system across all clusters. Each cluster maps its I/O routing paths through multiple "IO-Lanes". An "IO-Lane" is consisted of a group of I/O nodes managed and routed by an individual Gigabit/10gigabit Ethernet switch. Each IO-Lane provides accessibility to a set of storage subnets. With this we can support a Peta-scale global file system accessible for multi-cluster environments using the PaScalBB I/O architecture. We can linearly add more "IO- Lanes" into the PaScalBB I/O architecture to meet the increasing bandwidth demand of global parallel file systems. We could also have more than one switch per lane to make a lane fatter and for more fail over protection. The purpose of using IO Lane is to mitigate single switch bandwidth limitation and provide a linear growing path for IP storage network.

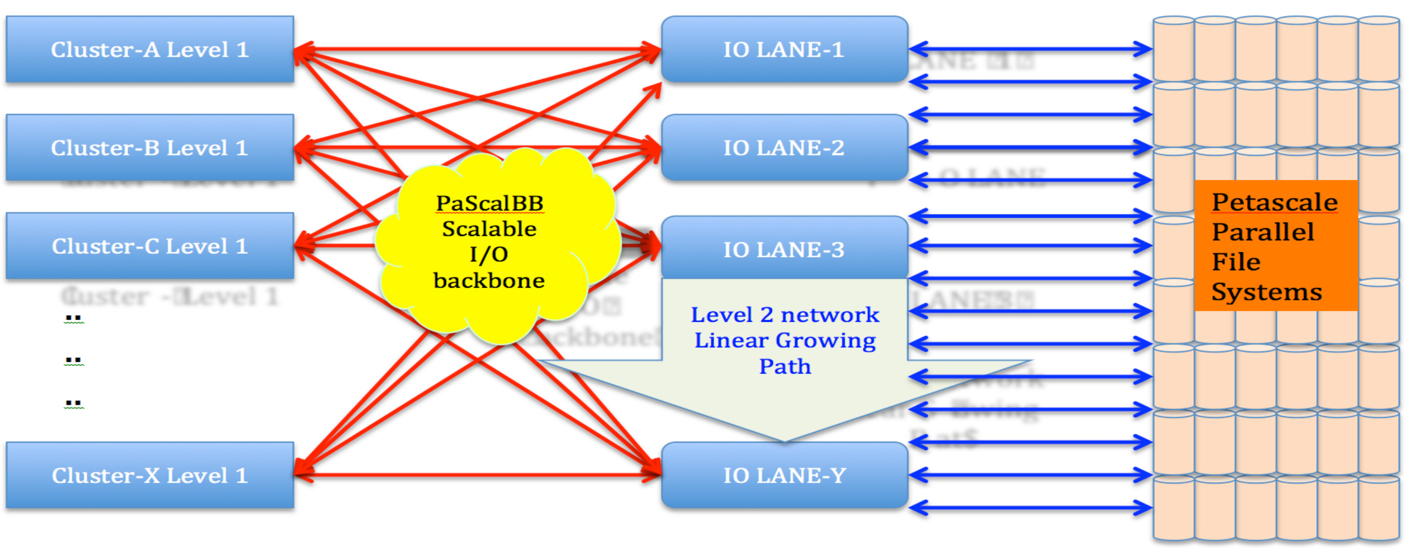


Figure-2: Multi-Clusters in PaScalBB: Supporting a scalable global Name space environment

How we built it, how it works

We have implemented the PaScalBB Server I/O network Infrastructure on several large-scale Linux Clusters (over 25000 nodes, 20+ large-size Linux clusters (Pink, Bolt, Lightning, Typhoon, Hurricane, TLCC-1, TLCC-2, Red-Tail, YellowRail, Turing, Cielto, RoadRunner, Cielo, Conejo, LightShow, Lobo, Luna, Mapache, Moonlight, Mustang, Pinto, Viewmaster, Wolf and two Petaflops computing cluster: RoadRunner and Cielo) at LANL in the past 10 years. PaScalBB has provided a smooth transition path from single Gigabit Ethernet network to 10 Gigabit Ethernet network. We were the first lab to have common scalable storage area network that connected muilti-clusters to single global parallel file system, we were also first to have supported automatic failure detection and reconfiguration in such a network as well. PaScalBB is very cost effective, because it adds bandwidth in the dimension where you need it, IO node to storage, but not where you already have bandwidth from node to node, unlike other storage area network designs that came along later.

We are currently migrating to 40-gigabit Ethernet I/O platform. Run-time experiences and historical statistics data have also shown that PaScalBB is very cost-effective solution in terms of administration and maintenance. It provides high throughput I/O Write/Read bandwidth in the range of 10GB/sec to 400GB/sec which depends on the scale and requirement of a cluster.

Next generation design of PaScalBB Server I/O infrastructure

Currently we are moving data sets in the 10s of TBs between tiered storage systems routinely. The increasing requirements of moving very large data sets from high performance computing simulation applications has led to the research of how to support greater I/O bandwidth and routing-efficient networking topologies. When we move to the Exascale computing era, Petabyte data sets will be common and will need to move data among tiered storage systems regularly.

With the advance of high speed interconnect technologies, such as Infiniband (FDR-56Gb/se and ERD-100Gb/sec) and Ethernet (40Gb/sec, 100Gb/sec), and processor technologies (many-core, multi-core), we are currently redesigning the PaScalBB infrastructure. We may adopt server and I/O virtualization technologies and leverage Software Defined Network (SND) capability so we can sustain more evolvable and flexible I/O networks.

Server virtualization is a technology for partitioning one physical I/O server into multiple virtual I/O servers. Each of these virtual I/O servers can run its own operating system and applications, and perform as if it is an individual I/O server. I/O Server virtualization would enhance the PaScalBB with (1) reduced total cost of ownership, (2) improved availability and operation continuity, and (3) increased efficiency for development and testing environment.

Software-defined networking (SDN) is an emerging paradigm that we can leverage in our PaScalBB infrastructure and provide more evolvable and flexible server I/O networks to meet the future Exascale computing I/O requirement. SDN supports the separation of data and control planes and employs a uniform vendor-agnostic interface. It can significantly reduce network administration and network management overhead by using a simple and uniform software program that manipulates the network flows within a logical slice of the network.

Where is PaScalBB going?

As stated above, PaScalBB is still our core HPC backbone connecting HPC clusters to our global Panasas file system, NFS servers and core resources such as Archive, NTP, LDAP and such. We are in the process of buying our first Advanced Technology machine named Trinity. This machine and other future machines will have 40Gige Ethernet interfaces that our core can't support today. Also our global production file systems are moving to Lustre File system. Lustre is generally provisioning using Infiniband (IB) as the data I/O interconnection network. This year we will be buying new Ethernet switches that will support

10/40/100gige. With Lustre being IB our HPC core network will become heterogeneous. We have named this project NGBB (Next generation BackBone). Like with PaScalBB we plan to design it with scalability and resilience as key features. Lustre's Lnet routers and bridge nodes will be used so older clusters can access the newer file systems. File transfer agents (FTAs) will still be used to transfer data to and from Scratch and Archive. However, our FTAs will now have to be more robust they will need multiple interfaces to attach to our heterogeneous core.

We believe as the Intel Corp. develops their new interconnect technology our FTAs or bridge nodes may have to connect to three different types of hardware media. We do believe we should keep fabric failure domains separate. Meaning the compute fabric will not be shared with the I/O fabric of the file system. We plan to extend the life of the HPC backbone longer than the life of clusters or file systems. Therefore there will be tradeoffs made in order to maintain a heterogeneous core. Tradeoffs may include but not limited to, some clusters having more bandwidth, some having better latency, or limited access to certain file systems. Networking is becoming more complex to adapt to changing hardware requirements from system vendors. It is challenging to design and buy something today that will be used for five or six years on systems that have not yet been built let alone designed by the vendors.

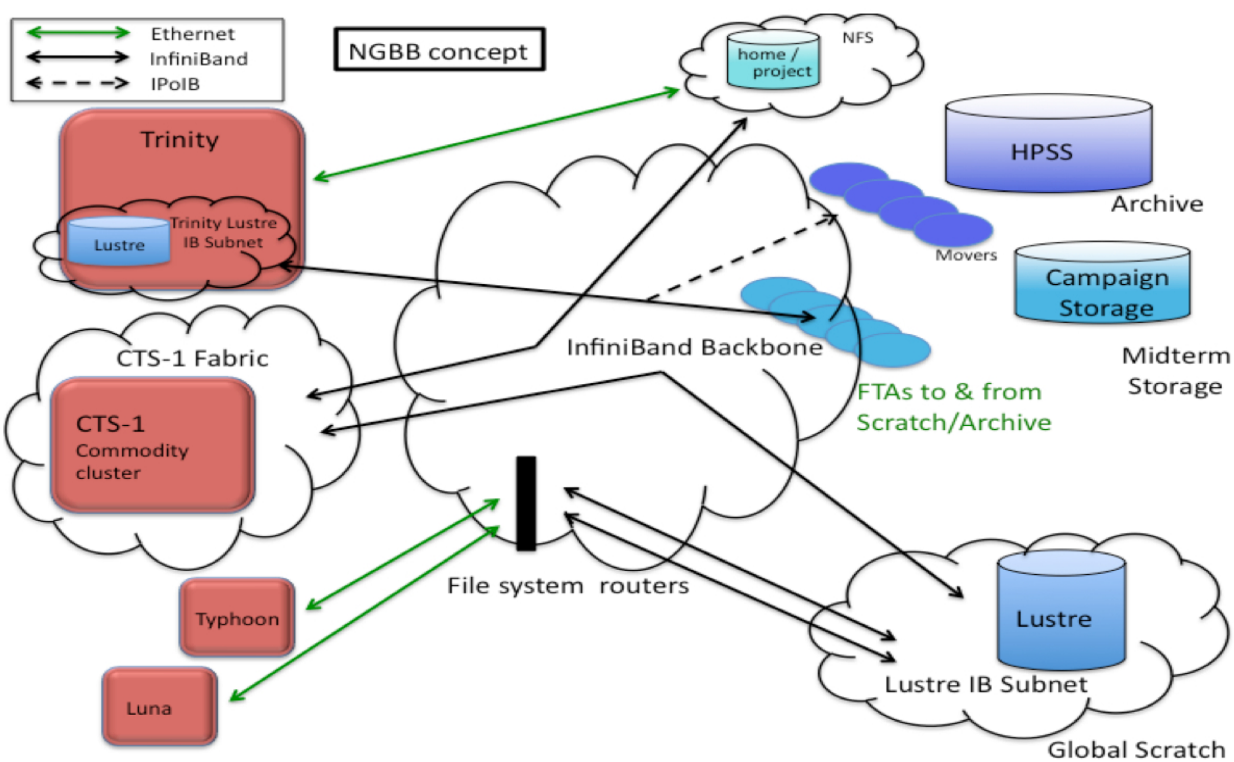
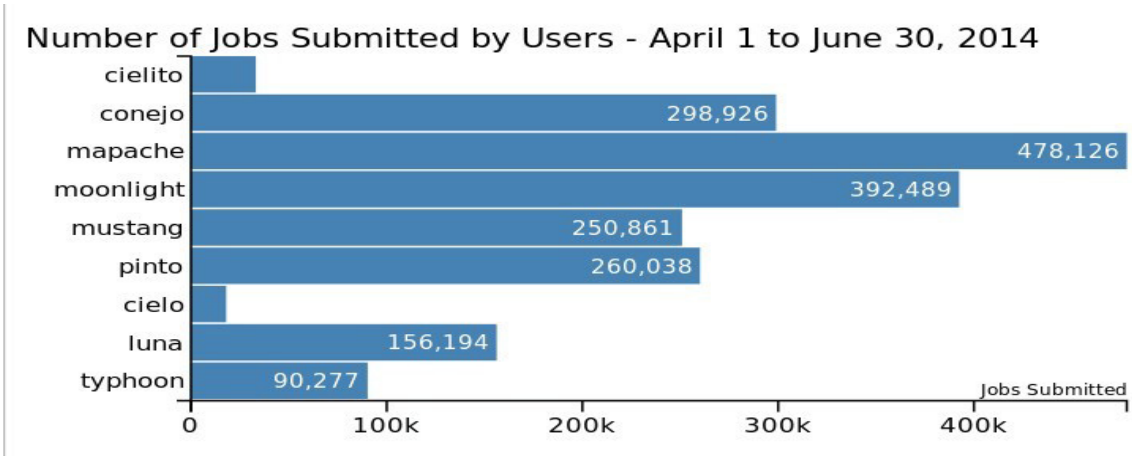
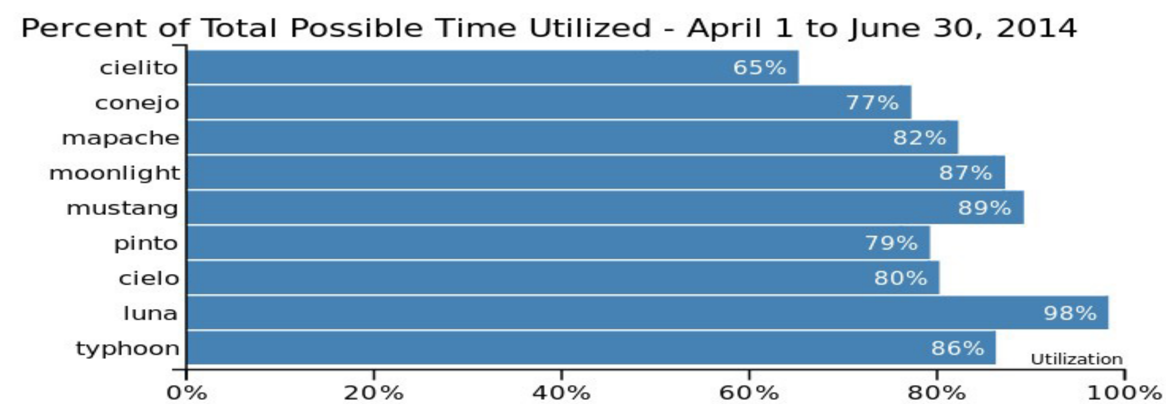
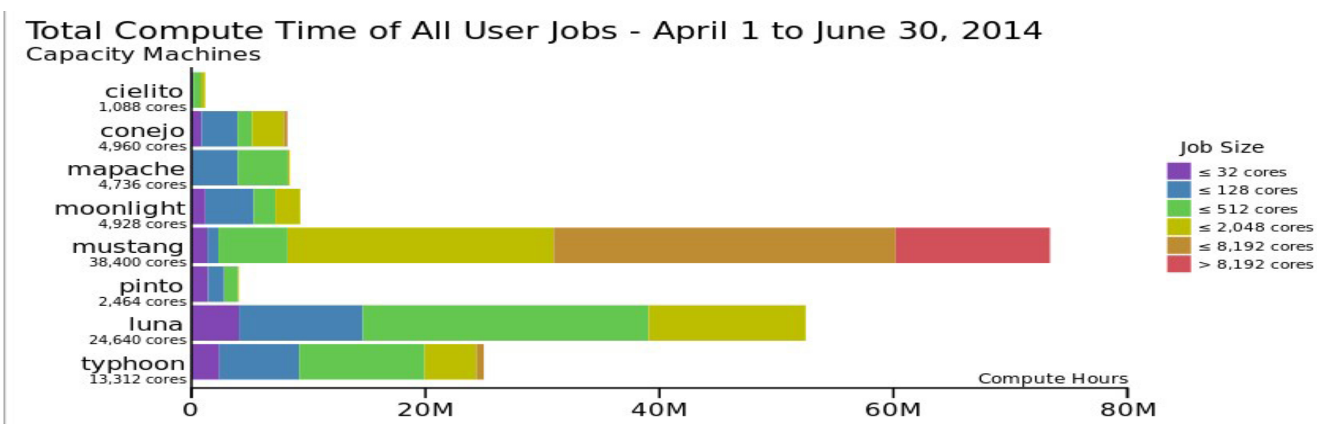


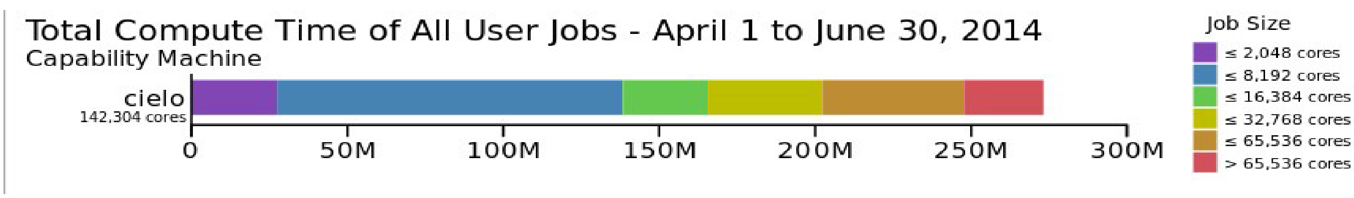
Figure-3: NGBB concept with FTAs shown

Quarterly Statistics









Current Machines- A snapshot in time

Name (Program[1])	Processor	OS	Total Compute Nodes	CPU cores per Node/ Total CPUs	Memory per compute Node/Total Memory	Interconnect	Peak (TFlop/s)	Storage
Secure Restricted Network (Red)								
Cielo (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	8,894 nodes	16/142,304	32 GB/297 TB[5]	3D Torus	1,370	10 PB Lustre
Luna TLCC2 (ASC)	Intel Xeon Sandybridge	Linux (Chaos)	1540 nodes	16/24,640	32 GB/49 TB	Qlogic InfiniBand Fat-Tree	513	3.7 PB Panasas
Typhoon (ASC)	AMD Magny-Cours	Linux (Chaos)	416 nodes	32/13,312	64 GB/26.6 TB	Voltaire InfiniBand Fat-Tree	106	3.7 PB Panasas
Open Collaborative Network (Turquoise)								
Cielito (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	68 nodes	16/1088	32 GB/2.3 TB[5]	3D Torus	10.4	344 TB Lustre
Conejo (LC)	Intel Xeon x5550	Linux (Chaos)	620 nodes	8/4960	24 GB/4.9 TB	Mellanox Infiniband Fat-Tree	52.8	1.8 PB Panasas
Lightshow[3] (ASC)	Intel Xeon	Linux (Chaos)	16 nodes	12/192	966 GB/1.5 TB	Mellanox Infiniband Fat-Tree	4.0	1.8 PB Panasas
Mapache (ASC)	Intel Xeon x5550	Linux (Chaos)	592 nodes	8/4736	24 GB/14.2 TB	Mellanox Infiniband Fat-Tree	50.4	1.8 PB Panasas
Moonlight TLCC2[3] (ASC)	Intel Xeon E5-2670 + NVida Tesla M2090	Linux (Chaos)	308 nodes	16/4,928 + GPUs	32 GB/9.86 TB	Qlogic Infiniband Fat-Tree	488	1.8 PB Panasas
Mustang (IC)	AMD Opteron 6176	Linux (Chaos)	1,600 nodes	24/38,400	64 GB/102 TB	Mellanox Infiniband at-Tree	353	1.8 PB Panasas
Pinto TLCC2[3] (IC)	Intel Xeon E5-2670	Linux (Chaos)	154 nodes	16/2464	32 GB/4.9 TB	Qlogic Infiniband Fat-Tree	51.3	1.8 PB Panasas
Wolf[2] TLCC2[3] (IC)	Intel Xeon E5-2670	Linux (Chaos)	616 nodes	16/9856	64 GB/39.4 TB	Qlogic Infiniband Fat-Tree	205	1.8 PB Panasas

[1] Programs: IC=Institutional Computing, ASC=Advanced Simulation and Computing, R=Recharge

[2] Wolf will become available Summer 2014

[3] TLCC = TriLab Linux Capacity Cluster; 2 = 2nd Generation

[5] Cielo has 372 viz nodes with 64GB memory each

[6] Cielito has 4 viz nodes with 64GB memory each